Designing, Operating, and Reoptimizing Elastic Optical Networks

Luis Velasco, Alba P. Vela, Fernando Morales, and Marc Ruiz

(Invited Tutorial)

***Abstract*—Emerging services and applications demanding high bitrate and stringent quality of service requirements are pushing telecom operators to upgrade their core networks based on wavelength-division multiplexing (WDM) to a more flexible technology for the more dynamic and variable traffic that is expected to be conveyed. Finally, academy- and industry-driven research on elastic optical networks (EON) has turned out into a mature enough technology ready to gradually upgrade WDM-based networks. Among others, key EON features include flexible spectrum allocation, connections beyond 100 Gb/s, advanced modulation formats, and elasticity against time-varying traffic. As a consequence of the variety of features involved, network design and algorithms for EONs are remarkably more complex than those for WDM networks. However, new opportunities for network operators to reduce costs arise by exploiting those features; in fact, the classical network life cycle based on fixed periodical planning cycles needs to be adapted to greatly reduce overprovisioning by applying reoptimization techniques to reconfigure the network while it is in operation and to efficiently manage new services, such as datacenter interconnection that will require provisioning multicast connections and elastic spectrum allocation for time-varying traffic. This paper reviews and extends mathematical models and algorithms to solve optimization problems related to the design, operation, and reoptimization of EONs. In addition, two use cases are presented as illustrative examples on how the network life cycle needs to be extended with in-operation planning and data analytics thus adding cognition to the network.**

***Index Terms*—Cognitive networking, elastic optical networks, in-operation planning, network planning.**

I. Introduction

ELASTIC optical networks (EON) [2] are now a reality and currently many network operators are in the process of migrating their transport networks by deploying optical nodes that include spectrum selective switches based on the flexgrid technology, as well as transponders (TPs) supporting advanced modulation formats such as the quadrature phase shift keying (QPSK) and the 16-ary quadrature-amplitude modulation (QAM16) [3]. In EONs, the available optical spectrum is divided into a set of frequency slices of a fixed spectral width (e.g., 6.25 GHz) and optical connections (lightpaths or light-trees) are assigned a number of slices according to their requested bitrate and the selected modulation format.

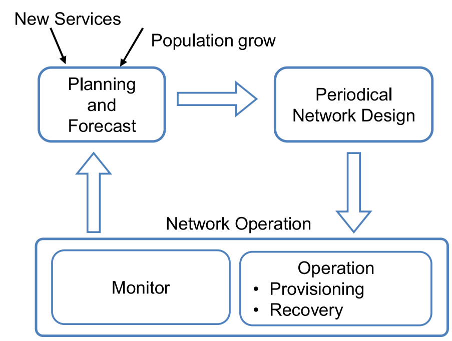


Fig. 1. Classical network life-cycle.

The finer spectrum granularity together with the technology supporting EON brings new features that cannot be offered by wavelength-division multiplexing (WDM) networks, such as: *i*) flexible spectrum allocation, allowing connections using a number of slices according to their need; *ii*) conveying connections with a capacity beyond 100 Gb/s; *iii*) improved spectral efficiency from using advanced modulation formats; and *iv*) elasticity against time-varying traffic i.e., lightpaths can elastically change their allocated spectrum to adapt their bitrate against spikes in the demand [4]. This is especially important for time-varying traffic, such as for datacenter interconnection [5], [6] and to deploy the telecom cloud [7].

Before deploying an EON, e.g., to gradually migrate from a WDM-based network [8], some activities need to be performed. In fact, the classical network life-cycle typically consists of several steps that are performed sequentially (see Fig. 1). Starting with inputs from the service layer and from the state of the resources in the already deployed network, a planning phase need to be carried out to produce recommendations that the next phase uses to design the network for a given period of time; network changes are verified and manually implemented.

Aiming at producing the input data required for the next planning cycle, network capacity is continuously monitored while it is in operation. Planning period is not fixed and its actual time length usually depends on many factors, which are operator and traffic type specific. Nonetheless, in case of unexpected in-

Manuscript received June 8, 2016; revised July 21, 2016; accepted July 21, 2016. Date of publication July 21, 2016; date of current version February 13, 2017. This work was presented in part at OFC 2016 [1]. This work was supported by the Spanish MINECO SYNERGY Project TEC2014-59995-R and by the Catalan Institution for Research and Advanced Studies.

The authors are with the Optical Communications Group, Universitat Politècnica de Catalunya, Barcelona 08034, Spain (e-mail: lvelasco@ac.upc.edu; apvela@ac.upc.edu; fmorales@ac.upc.edu; mruiz@ac.upc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2016.2593986

creases in demand or network changes detected by monitoring, the planning process may be restarted.

Designing and dimensioning the network, generally consists in determining the nodes and links that need to be installed and which is the equipment to be purchased to serve the foreseen traffic while minimizing network capital expenditures (CAPEX). To do so, traffic demands need to be routed over the network and a portion of the optical spectrum allocated for each of them and therefore, the routing, modulation and spectrum allocation (RMSA) problem is part of the optimization problem that needs to be solved.

Network operation requires solving the RMSA problem for one (provisioning) or a small set of demands (recovery). Note that in contrast to off-line planning problems, operation problems require stringent solving times.

The classical life-cycle can be extended by introducing network reconfiguration so as to optimize resources utilization while the network is in-operation; this was named as *in-operation planning* in [9]. Network reconfiguration requires solving optimization problems, where the obtained solutions are immediately implemented in the network. Note that this is opposite to traditional network planning, where results and recommendations require from manual intervention and hence long time is required until they are implemented in the network. In-operation planning might involve solving the RMSA problem.

In summary, network planning problems can be solved off-line since the network is not yet in operation and hence, no limit in the time to solve those problems is generally required. Conversely, when the network is in operation, stringent solving times are usually required. Focusing on both, design, operation and in-operation planning, the contribution of this paper is three-fold:

1) Firstly, Sections II–IV review and extend previous works related to design, operation and re-optimization of EONs. They are presented in a consistent and integrated way to facilitate readers understanding both modelling and solving planning problems. Section II presents integer linear programming (ILP) formulations involving the RMSA problem for lightpaths and light-trees. Section III focuses on dynamic provisioning and presents heuristic algorithms for single lightpath and light-tree provisioning, as well as for bitrate adaptation. Finally, Section IV concentrates on in-operation network re-optimization, specifically on the defragmentation problem and presents alternative algorithms to be applied for provisioning and for elastic spectrum operations;
2) as a use case where RMSA needs to be solved for a set of demands, a recovery problem and an in-operation problem to reconfigure the network after failure repair are introduced, formally stated and algorithms are proposed;
3) a second use case for network reconfiguration is proposed in multi-layer scenarios. In such scenarios, unexpected increment of traffic (*traffic anomaly*) might create congestion in the network if it is not promptly detected and the *virtual network topology* (VNT) reconfigured to be able to convey the traffic increment.

TABLE I
CHARACTERISTICS OF DIFFERENT MODULATION FORMATS

Modulation Format	Spectral Efficiency(bit/symbol)	Reach (km)
DP-QPSK	4	3000
DP-QAM8	6	1800
DP-QAM16	8	900

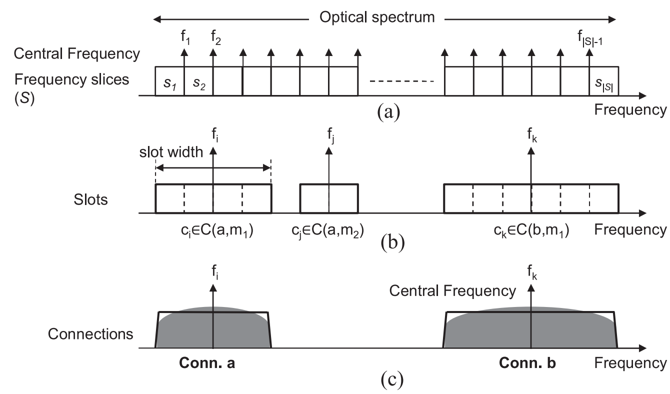


Fig. 2. Basic concepts used in this paper. (a) Optical spectrum is divided into frequency slices. (b) Lightpaths use a number of contiguous slices, named as slot. (c) Slot assignment.

II. OFF-LINE PLANNING PROBLEMS

In this section we introduce some basic concepts related to the RMSA problem and review off-line planning problems solving that problem for point-to-point (*lightpaths*) and point-to-multipoint (P2MP) (*light-trees*) optical connections.

A. Concepts for Modelling the RMSA Problem

Before start, it is worth noting that different modulation formats might provide different spectral efficiencies, which translate into different spectral width. For instance, an optical connection requesting 400 Gb/s would need 50 GHz if the DP-QAM16 modulation format is selected in contrast to 200 GHz needed if the DP-QPSK modulation format is used. However, the reachability of more efficient modulation formats is shorter that those less efficient and therefore, the selection of a modulation format depends on the length of the selected path. Table I presents illustrative values for the spectral efficiency and reachability of several modulation formats using dual polarization (DP).

The RMSA problem consists in finding a feasible route, modulation format, and a spectrum allocation for a set of demands. Two constraints related to the spectrum allocation need to be enforced: *i*) the spectrum continuity constraint. In flexgrid, the spectrum allocation is represented by a *slot* and thus, in the absence of spectrum converters, the same slot must be used along the links of a given routing path; *ii*) the spectrum contiguity constraint, where the allocated *slices* must be contiguous in the spectrum.

For illustrative purposes, Fig. 2(a) shows an example of the optical spectrum divided into frequency slices, each of the same width (e.g., 6.25 GHz). A number of frequency slices (frequency slot) are allocated to every connection around a central frequency. Note that the spectrum width of each slot depends on the bitrate requested and the modulation format (see Fig. 2(b)). A spectrum allocation for a connection request entails thus, selecting a specific slot (width and central frequency) and a modulation format (see Fig. 2(c)).

Although several works can be found in the literature presenting ILP formulations for the simpler RSA problem, in this paper we rely on those in [10] since their approach, based on the assignment of slots, allows efficiently solving that problem.

The definition of slots can be mathematically formulated as follows. Let us assume that a set of slots *C*(*d*) is pre-defined for each demand *d* that requests b_d bitrate. Depending on the modulation format *m* selected among the available set *M*, the size of the slot would be different, so $C(d) = \cup_{m \in M} C(d,m)$, where every slot in $C(d,m)$ contains the same number n_{dm} of slices. For convenience, we define q_{cm} equal to 1 if $c \in C(d,m)$.

Let q_{cs} be a coincidence coefficient equal to 1 whenever slot $c \in C$ uses slice $s \in S$, and 0 otherwise. Hence, $\forall c \in C(d,m)$ the spectrum contiguity constraint is implicitly imposed by the proper definition of q_{cs} such that $\forall i, j \in S : q_{ci} = q_{cj} = 1, i < j \Rightarrow q_{ck} = 1, \forall k \in \{i, \ldots, j\}, \sum_{s \in S} q_{cs} = n_{dm}$. In this paper, we consider that each set $C(d,m)$ consists of all possible slots of size n_{dm} that can be defined in *S*. Since $|C(n_{(\cdot)})| = |S| - (n_{(\cdot)} - 1)$, the size of the complete set of slots *C* that needs to be defined is $|C| = \sum_{d \in D} \sum_{m \in M} [|S| - n_{dm} + 1] < |D| \cdot |M| \cdot |S|$.

Therefore, we can define the RMSA problem as the problem that finds a proper lightpath, i.e., a route, modulation format, and slot, for each demand from a given set so that the number of active slices in the assigned slot guarantees that the bitrate requested by each demand can be transported. Note that by precomputing the set of slots that can be assigned to each demand, the complexity added by the contiguity constraint is removed.

B. Basic RMSA Problem

A very basic RMSA problem consists in finding a lightpath represented by tuple the $\langle p, m, c \rangle$, containing a path *p*, a modulation format *m*, and a frequency slot *c*, for every demand in a given traffic matrix with the objective of minimizing or maximizing some utility function. Several alternatives for this problem may exist, for instance we can assume that all the traffic matrix need to be served, or alternatively some demands can be blocked, i.e. not served. The problem can be formally stated as follows.

Given:

1) a connected directed graph $\mathrm{G}(N, E)$, where *N* is the set of locations and a *E* is the set of optical fibers connecting two locations,
2) the optical spectrum width *S* and the set of modulation formats *M*,
3) a traffic matrix *D* with the amount of bitrate exchanged between every pair of locations in *N*.

Output: the lightpath $\langle p, m, c \rangle$ for each demand in *D*.

Objective: one or more among:

1) Minimize the amount of bitrate blocked,
2) Minimize the total amount of used slices,
3) etc.

In the following, we present an ILP model for the above problem, based on the formulations in [10]. Note that since the topology is given, we can pre-compute a set of *k* distinct paths for every demand in the traffic matrix and hence, the formulation is usually known as *arc-path*. Moreover, because of the use of pre-computed modulation format aware slots for each demand, we call this formulation as arc-path routing modulation and slot assignment (AP-RMSA).

The following sets and parameters have been defined.

Topology:

- *N* Set of locations, index *n*.
- *E* Set of fiber links, index *e*. The length len(*e*) for every link is defined.

Demands and Paths:

- *D* Set of demands, index *d*. For each demand *d*, the tuple $\langle o_d, t_d, b_d \rangle$ is given, where o_d and t_d are the origin and target nodes, and b_d is the requested bitrate in Gb/s.
- *P* Set of pre-computed paths, index *p*.
- *P*(*d*) Subset of pre-computed paths for demand *d*. $|P(d)| \leq k \forall d \in D$
- r_{pe} Equal to 1 if path *p* uses link *e*.
- len(*p*) Length of path *p* computed as $\sum_{e \in E} r_{pe} \cdot \mathrm{len}(e)$.

Spectrum and Modulation Formats:

- *S* Set of spectrum slices, index *s*.
- *C*(*d*) Set of pre-computed slots for demand *d*.
- *M* Set of modulation formats, index *m*. The reachability len(*m*) for every modulation format is defined.
- q_{cm} Equal to 1 if slot $c \in C(d)$ is computed for modulation format *m*.
- q_{cs} Equal to 1 if slot *c* uses slice *s*.

The Decision Variables are:

- w_d Binary, equal to 1 if demand *d* cannot be served.
- x_{dpc} Binary, equal to 1 if demand *d* is routed through path *p* and slot *c*.

The AP-RMSA formulation is as follows:

$$(AP - RMSA) \ \min \sum_{d \in D} b_d \cdot w_d \tag{1}$$

subject to:

$$\sum_{p \in P(d)} \sum_{c \in C(d)} x_{dpc} + w_d = 1, \ \forall d \in D \tag{2}$$

$$\sum_{p \in P(d)} \sum_{c \in C(d)} q_{cm} \cdot len(p) \cdot x_{dpc} \leq len(m), \quad \begin{matrix} \forall d \in D, \\ m \in M \end{matrix} \tag{3}$$

$$\sum_{d \in D} \sum_{p \in P(d)} \sum_{c \in C(d)} r_{pe} \cdot q_{cs} \cdot x_{dpc} \leq 1, \forall e \in E, s \in S. \tag{4}$$

The objective function (1) minimizes the amount of bitrate that cannot be served (rejected). Constraint (2) ensures that a lightpath is selected for each demand provided that the demand is served; otherwise the demand cannot be served and therefore, is rejected. Constraint (3) selects a modulation format with enough reachability for the selected path. Finally, constraint (4) guarantees that every slice in every link is assigned to one demand at the most.

The size of the AP-RMSA formulation is $\mathrm{O}(k \cdot |D| \cdot |C|)$ variables and $\mathrm{O}(|E| \cdot |S| + |D| \cdot |M|)$ constraints. As an example, the size of the above formulation for a typical European national network with 22 nodes and 35 links, considering $|S| = 80, |M| = 3, |D| = 100$, and $k = 10$, is $8 \cdot 10^4$ variables and $3 \cdot 10^3$ constraints, which is a noteworthy high even for this very simple problem.

The AP-RMSA formulation can be classified as non-compact since it requires from an exponential number of x_{dpc} variables for the whole set of possible lightpaths. However, only few of them will be non-zero in the optimal solution. Based on this fact, authors in [11] describe column generation solving techniques to generate reduced sets of variables providing high quality solutions that can be easily applied when the link-path formulation is used. Moreover, large scale optimization techniques such as those presented in [12] can be applied to solve the link-path formulation.

C. Topology Design as an RMSA Problem

In the previous RMSA problem the network topology was given. Let us consider now that the problem consists in designing the network topology to serve all the demands in the given traffic matrix. Since each installed link increases network CAPEX as a result of optical interfaces, including amplifiers, to be installed in the end nodes and some intermediate locations, minimizing the number of links in the resulting network topology would reduce total CAPEX cost.

The problem can be formally stated as follows:

Given:

1) a connected directed graph $\mathrm{G}(N, E)$,
2) the optical spectrum width *S* and the set of modulation formats *M*,
3) a traffic matrix *D*.

Output: the lightpath $\langle p, m, c\rangle$ for each demand in *D* and the links to be equipped.

Objective: Minimize the number of links to be equipped to transport the given traffic matrix.

Note that we could pre-compute *k* distinct paths for each demand in the traffic matrix, as we did in the previous problem. However, since only part of the links will be eventually installed, the number of routes *k* to be pre-computed for each demand would need to be highly increased to counteract the fact that some of the routes would become useless. For that very reason, we present an ILP formulation named as *node-arc* that performs routing computation within the optimization. Similarly as before, since we pre-compute modulation format –aware slots for each demand, we call this formulation as node-arc modulation format and slot assignment (NA-RMSA).

New Sets and Parameters Have Been Defined:

$E^+(n)$	Subset of *E* with links arriving node *n*.
$E^-(n)$	Subset of *E* with links leaving node *n*.
g_{ne}	Equal to 1 if link *e* is incident to node *n*.

The Decision Variables are:

x_{dec}	Binary, equal to 1 if demand *d* uses slot *c* in link *e*.
y_{dc}	Binary, equal to 1 if demand *d* uses slot *c*.
z_e	Binary, if link *e* is installed.

The NA-RMSA formulation is as follows:

$$(NA - RMSA) \min \sum_{e\in E} z_e \tag{5}$$

subject to:

$$\sum_{e\in E^+(n)} \sum_{c\in C(d)} x_{dec} - \sum_{e\in E^-(n)} \sum_{c\in C(d)} x_{dec} = \begin{cases} 1 & \forall d \in D, n = o_d \\ 0 & \forall d \in D,\ n \in N \\ -1 & \forall d \in D, n = t_d \end{cases} \tag{6}$$

$$\sum_{e\in E} x_{dec} \le |E| \cdot y_{dc} \ \ \forall d \in D, c \in C \tag{7}$$

$$\sum_{c\in C} y_{dc} = 1 \ \forall d \in D \tag{8}$$

$$\sum_{e\in E} \sum_{c\in C(d)} q_{cm} \cdot len(e) \cdot x_{dec} \le len(m) \ \ \forall d \in D, m \in M \tag{9}$$

$$\sum_{d\in D} \sum_{c\in C(d)} q_{cs} \cdot x_{dec} \le z_e,\ \forall e \in E, s \in S. \tag{10}$$

The objective function (5) minimizes the amount of links to be installed. Constraints (6) find a lightpath for every demand. Specifically, it ensures that one lightpath for each demand leaves from the source node and enters in the target node of every demand, whereas guarantees that every lightpath entering an intermediate node for a demand, leaves that node. Note that solutions with cycles can be produced since the length of paths is not strictly minimized and thus, a post-processing phase needs to be applied to remove them (if any) with no impact on feasibility and optimality of solution.

Constraints (7) and (8) guarantee that every lightpath is assigned one and only one slot along its route; constraint (7) collects the slots that are assigned to every lightpath whereas constraint (8) limits that number to one. Similarly to constraint (3) in the AP-RMSA formulation, constraint (9) selects a modulation format with enough reachability for the selected path. Finally, constraint (10) prevents that any frequency slice in any link is used by more than one demand, while installing the link when any slice is used.

The size of the NA-RMSA formulation is $\mathrm{O}(|D| \cdot |E| \cdot |C|)$ variables and $\mathrm{O}(|E| \cdot |S| + |D| \cdot (|N| + |M| + |C|))$ constraints. The size of this formulation for the national network previously considered is $2.8 \cdot 10^5$ variables and $1.3 \cdot 10^4$ constraints, higher than the AP-RMSA formulation.

It is obvious that minimizing the number of links to be installed can be different than minimizing CAPEX, since some other costs need to be considered. For this very reason, the previous problem needs to be extended to take into account all the costs and to dimension every equipment in the network.

Then, a more specific network dimensioning problem can be formally stated as follows:

Given:

1) a connected graph G(*N*, *E*),
2) the optical spectrum width and the set of modulation formats,
3) a traffic matrix *D*,

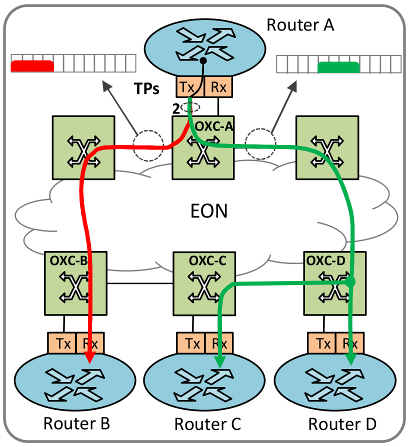


Fig. 3. Optical multicast provisioning, where two transparent light-trees (one of them is just a path) serve together a single multicast demand.

4) the cost of every component, such as optical cross-connects (OXC), TP types and regenerators specifying its capacity and reach. The cost of installing each link is also specified.

Output:
1) the lightpath $\langle p, m, c\rangle$ for each demand in *D*,
2) the links that need to be equipped and the network dimensioning including the type of OXC, TPs, and regenerators in each location;

Objective: Minimize the total CAPEX to transport the given traffic matrix.

Although in this paper we do not present any specific ILP model for this problem, it could be easily derived from the NA-RMSA formulation together with a CAPEX model (see e.g., [13], [14]).

D. Topology Design for Optical Multicast

Although EONs usually serve point-to-point connections, P2MP connections can also be served, e.g., for datacenter interconnection. A multicast demand *d* can be defined by the tuple $\langle o_d, T(d), b_d\rangle$, where *T*(*d*) represents the set of destination nodes.

Fig. 3 illustrates the routing scheme proposed by authors in [15] to serve multicast demands in the optical layer, where client routers/switches are connected to OXCs by means of 100 Gb/s optical TPs. In this scenario, the multicast demand ⟨A, {B, C, D}, 100 Gb/s⟩ can be served by one single light-tree. In such case, one single transmission (Tx) TP needs to be used in the source node. However, since transparent trees require one single modulation format and frequency slot, its performance might drop when long source-to-destination routes are required. In light of that fact, that constraint can be relaxed and hence, a set of transparent light-trees can be used to serve a single multicast demand. The drawback of this scheme is that increases TP count in the source node, so the number of light-trees serving every multicast demand should be minimized.

The P2MP-RMSA problem can be formally stated as:

Given:
1) a connected graph $G(N, E)$,
3) the optical spectrum width *S* and the set of modulation formats *M*,
3) a set *D* of multicast demands to be served, where each demand *d* is identified by the tuple $\langle o_d, T(d), b_d\rangle$.

Output: the set of light-trees for each demand in *D*, where every light-tree is defined by the tuple $\langle p, m, c\rangle$, being *p* a tree.

Objective: Minimize the number of links to be equipped to transport the given traffic matrix (main objective) and the number of light-trees (secondary objective).

Note that the secondary objective is especially important to minimize the number of Tx TPs in the source node of the demands. As a result, a single light-tree for each demand will be selected unless a single spectrum allocation cannot be found for such demand.

The following ILP is based on the NA-RMSA formulation described above. Routing constraints build directed Steiner trees [16], while spectrum allocation is performed by assigning pre-computed slots to each computed Steiner tree. To obtain directed Steiner trees, one lightpath is found for each sub-connection, i.e., origin-destination pair, and those paths sharing links and slots are merged into a light-tree.

In Addition to the Previous Notation, the Following to the Set and Parameter are (Re)Defined:

D	Set of multicast demands, index *d*. Each demand *d* is defined by the tuple $\langle o_d, T(d), b_d\rangle$.
α	Cost multiplier.

The Next Decision Variables are Also Defined:

w_{dc}	Binary, equal to 1 if slot *c* is assigned to demand *d*; 0 otherwise.
x_{dtec}	Binary, equal to 1 if destination *t* of demand *d* is reached through link *e* and slot *c*; 0 otherwise.
y_{dtc}	Binary, equal to 1 if destination *t* of demand *d* is reached through slot *c*; 0 otherwise.

The P2MP-RSA formulation is as follows:

$$(P2MP - RMSA) \min \sum_{e\in E} z_e + \alpha \cdot \sum_{d\in D} \sum_{c\in C(d)} w_{dc} \quad (11)$$

subject to:

$$\sum_{e\in E^+(n)} \sum_{c\in C(d)} x_{dtec} - \sum_{e\in E^-(n)} \sum_{c\in C(d)} x_{dtec} = \begin{cases} 1 & \forall d \in D, t \in T(d), \\ & n = o_d \\ 0 & \forall d \in D, t \in T(d), \\ & n \in N \backslash \{o_d, t\} \\ -1 & \forall d \in D, t \in T(d), \\ & n = t \end{cases} \quad (12)$$

$$\sum_{e\in E} x_{dtec} \le |E| \cdot y_{dtc} \quad \forall d \in D, t \in T(d), c \in C \quad (13)$$

$$\sum_{c\in C} y_{dtc} = 1 \quad \forall d \in D, t \in T(d) \quad (14)$$

$$\sum_{e\in E} \sum_{c\in C(d)} len(e) \cdot x_{dtec} \le len(m), \quad \forall d \in D, t \in T(d), m \in M \quad (15)$$

$$\sum_{t\in T(d)} x_{dtec} \le |T(d)| \cdot x_{dec}, \quad \begin{array}{l}\forall d \in D, e \in E,\\ c \in C(d)\end{array} \tag{16}$$

$$\sum_{e\in E} x_{dec} \le |E| \cdot w_{dc}, \ \forall d \in D, c \in C(d) \tag{17}$$

$$\sum_{d\in D} \sum_{c\in C(d)} q_{cs} \cdot x_{dec} \le z_e, \ \forall e \in E, s \in S. \tag{18}$$

The objective function (11) minimizes both the amount of links to be installed and the number of light-trees created (used Tx TPs). Constraints (12) to (15) deal with sub-connection routing using a continuous slot and a valid modulation format and are similar to constraints (6) to (9) in the NA-RMSA formulation. Constraint (16) is in charge of building light-trees by joining sub-paths of the same demand sharing the same slot in one or more links. Constraint (17) stores the number of distinct slots used for each demand. Finally, constraint (18), identical to constraint (10), makes sure that slices are assigned to one connection at the most.

Note that the size of the P2MP-RSA formulation is $O(|D| \cdot |T(d)| \cdot |E| \cdot |C|)$ variables and $O(|D| \cdot (|E| \cdot |S| + |T(d)| \cdot (|N| + |M| + |C|)))$ constraints, slightly higher that the NA-RMSA formulation since it entails several connections for every demand.

III. Dynamic Provisioning

In contrast to the problems presented in Section II, in this section we focus on those involving the RMSA problem for a single point-to-point or P2MP connection request to be served. Note that this time the network is in operation, so some connections are already established using some of the network resources. In such case, time to solve is very important so we review heuristic algorithms providing good trade-off between optimality and complexity.

A. RMSA Algorithm for Single Demand Provisioning

Let us analyze the special case where the RMSA problem needs to be solved for a single demand that can be stated as follows:

Given:

1) a connected graph $G(N, E)$;
2) the availability of every frequency slice in the optical spectrum of every link in *E*;
3) a set *M* of modulation formats, where *M* is ordered by its spectral efficiency;
4) a connection request to be served defined by the tuple $\langle o, t, b\rangle$.

Output: the lightpath $\langle p, m, c\rangle$ for the request.

Objective: serve the connection request while minimizing the used resources.

Authors in [17] included the selection of the TP in the optimization problem as a result of the particular TP architecture they proposed; however, in general that selection can be performed once a feasible lightpath $\langle p, m, c\rangle$ has been found.

Authors in [18] proposed the algorithm presented in Table II to solve the RMSA problem for a single demand. A set of

TABLE II
RMSA ALGORITHM

INPUT: $G(N, E)$, $\langle o, t, b\rangle$, *minSlotWidth*
OUTPUT: $\langle p, m, c\rangle$

1: $Q = \{< p, n >\} \leftarrow \mathrm{kSP}(G, o, t)$
2: $Q' \leftarrow \varnothing$
3: **for each** q **in** Q **do**
4:   **if** $q.n < minSlotWidth$ **then continue**
5:   $q.m \leftarrow 0$
6:   **for each** m **in** M **do**
7:     **if** $\mathrm{len}(q.p) \le \mathrm{len}(m)$ AND width $(b, m) \le q.n$ **then**
8:       $q.m \leftarrow m$
9:       **break**
10:   **if** $q.m \ne 0$ **then** $Q' \leftarrow Q' \cup \{q\}$
11: **if** $Q' \leftarrow \varnothing$ **then return** Ø
12: sort(Q', $|q.p|$, ASC)
13: $q \leftarrow$ first(Q')
14: $q.c \leftarrow$ selectSlot(p, width(b, m))
15: **return** q

TABLE III
Elastic Spectrum Allocation Algorithm

INPUT: $G(N, E), L, l = \langle p, m, f, n\rangle, b_{req}$
OUTPUT: $\langle f', n'\rangle$

$n_{req} \leftarrow$ width$(b_{req}, l.m)$
if $n_{req} \le l.n$ **then return** $\langle l.f, n_{req}\rangle$
$L^+ \leftarrow \varnothing, L^- \leftarrow \varnothing$
for each $e \in l.p$ **do**
  $L^- \leftarrow L^- U\{l' \in L : e \in l'.p,$ adjacents $(l, l'), l'.f < l.f\}$
  $L^+ \leftarrow L^+ U\{l' \in L : e \in l'.p,$ adjacents $(l, l'), l'.f < l.f\}$
$n_{max} \leftarrow \min\{l.f - l'.f - l'.n, l' \in L^-\}$
    $+ \min\{l'.f - l.f - l'.n, l' \in L^+\}$
$n' \leftarrow \min\{n_{max}, n_{req}\}$
if $n = l'.n$ **then return** $\langle l.f, l.n\rangle$
$f' \leftarrow$ findSA_MinCFShifting $(G, l.p, n', L^+, L^-)$
return $< f', n' >$

shortest paths are first computed between end nodes (line 1 in Table II); each path includes its physical route *p* and the width of the largest continuous slot in that route, n_p. Each path is afterwards checked to verify the width of largest slot available (line 4). Next, the best modulation format is selected from set *M* provided that the reach works for the length of the route (lines 6–9). If no path satisfies the previous constraints, the connection request is blocked (line 11). Otherwise, the set of found paths is sorted first by the length of its route and the best path is selected (lines 12–13). A slot of the proper width is selected (line 14) and the computed lightpath is eventually returned.

The complexity of the RMSA algorithm strongly depends on that of the selected kSP algorithm; in this work, we propose using the Yen's kSP algorithm with complexity $O(k \cdot |N| \cdot (|E| + |N| \cdot \log|N|))$ [19].

B. Elastic Spectrum Allocation for Time-Varying Traffic

A different case is when the capacity of an existing lightpath needs to be increased or decreased to follow time-varying traffic [4]. Authors in [20] explored the elastic spectrum allocation capability of EONs and proposed algorithms to adapt the spectrum allocated to lightpaths in response to traffic changes. The problem of dynamic lightpath adaptation can be formally stated as follows. Note that for the sake of clarity, in this problem frequency slots are defined as tuples $\langle f, n\rangle$ specifying their central frequency (*f*) represents the slot width in terms of number of slices (*n*).

Given:

1) a connected graph $\mathrm{G}(N, E)$;
2) the availability of every frequency slice in the optical spectrum of every link in *E*;
3) a set *L* of lightpaths already established on the network, where each *l* is defined by the tuple $\langle p_l, m_l, f_l, n_l\rangle$;
4) a lightpath *l* in *L* for which a bitrate adaptation request b_{req} arrives.

Output: the new values for the spectrum allocation of the given lightpath $\langle f', n'\rangle$.

Objective: minimize first the amount of unserved bitrate and then, minimize spectrum shifting.

We assume that the central frequency of established lightpaths can be shifted without traffic disruption; to that end, the *push-pull* technique [21] can be used. Push-pull consists in re-tuning the transmitter laser from the original to the target nominal central frequency, while the receiver is automatically pulled to track the signal shifting. The limitation of the central frequency shifting performed using the push-pull technique resides on the fact that lightpaths can be allocated to new slots as long as no other lightpaths are established between the current and the new slot, reducing thus the set of reallocations available for each lightpath.

Table III reproduces the algorithm. The requested slot width (n_{req}) is first computed and in case that elastic spectrum reduction is requested, the slot width is reduced while the central frequency is kept invariant (line 2). In the opposite, when an elastic expansion is requested, the set of spectrum adjacent lightpaths at each of the spectrum sides is found by iterating on the links served on any of the links in *p* (lines 4–6). The largest slot width to be allocated to the lightpath (n_{max}) is constrained by the amount of slices available between the closest spectrum-adjacent paths. n_{max} is computed as the sum of the minimum available amount of slices along the links in the left side of the spectrum and the minimum available slices in the right side of the allocated spectrum of a given lightpath (line 7). *n*' is computed as the minimum between n_{max} and n_{req} and the *f*' is obtained by computing the new central frequency that minimizes frequency shifting (lines 8–10).

Note that the complexity of the algorithm is proportional to the length of the route of the lightpath plus that of the algorithm to select the new central frequency.

C. Multicast Provisioning Problem

Finally, let us analyze the case of P2MP connection provisioning, which can be stated as follows:

Given:

1) a connected graph G(*N*, *E*);
2) the availability of every frequency slice in the optical spectrum of every link in *E*;
3) a set *M* of modulation formats, where *M* is ordered by its spectral efficiency;

TABLE IV
LIGHT-TREE PROVISIONING ALGORITHM

INPUT $\mathrm{G}(N, E), \langle o, T, b\rangle$
OUTPUT $\{\langle p, m, c\rangle\}$

1: $Q = \{\langle p, m, C\rangle\} \leftarrow <\varnothing, 0$, computeAllSlots(*b*,*M*)>
2: **while** $T \neq \varnothing$ **do**
3: $maxCost \leftarrow -\infty$
4: $Q* \leftarrow \varnothing$
5: **for each** $t \in T$ **do**
6: $<Q',\ cost> \leftarrow$ RMSel (kSP(G,*o*,*t*), *T*, *Q*, *b*)
7: **if** *cost* > *maxCost* **then**
8: *maxCost* ← *cost*
9: $Q* \leftarrow Q'$
10: **if** $Q* = \varnothing$ **then return** INFEASIBLE
11: $Q \leftarrow Q*$
12: *T*←updateDestinations(*T*, *Q*)
13: **for each** $q \in Q$ **do** *q.c*←selectSlot(*q.C*)
14: **return** $Q = \{\langle p, m, c\rangle\}$

4) a connection request to be served defined by the tuple $\langle o, T, b\rangle$.

Output: the set of light-trees to serve the connection request, where every light-tree is defined by the tuple $\langle p, m, c\rangle$, being *p* a tree.

Objective: serve the connection request while minimizing the used resources and the number of light-trees.

Authors in [22] proposed algorithms for multicast provisioning experimentally validated later in [23]. However, those algorithms do not include modulation format selection and were designed to provision a single light-tree for each multicast demand. The algorithm in Table IV extends those previous algorithms; it consists in a routing and modulation format selection phase to build a set of light-trees followed by a spectrum allocation phase. Although both phases are independent, the routing and modulation format selection phase guarantees the availability of at least one continuous frequency slot for each light-tree.

The algorithm starts initializing a solution with a void tree containing all possible slots for the connection request (line 1 in Table IV). Next, an iterative procedure finds paths from source to destinations and merges into a set of light-trees (lines 2–12). The procedure runs until all the destinations are added to one of the light-trees. At each iteration, remaining destinations are evaluated as candidates to be inserted into the current solution, by either extending one of the existing light-trees or creating a new one. The path with the minimum cost for each remaining destination is found (line 6). Among all destinations, the one with the highest minimum cost is selected (lines 7–9). The solution is infeasible if no path is available for some destinations (line 10). On the contrary, the solution is updated (line 11) and those destinations already visited removed from the set *T* (line 12).

After the routing phase, *Q* contains a set of trees each connecting the source with a subset of destination nodes, one modulation format, and a set of end-to-end available slots along the tree. Then, one of the available slots is selected (line 13) and the solution is eventually returned.

The light-tree provisioning algorithm in Table IV relies on the RMSel algorithm in Table V that aims at extending a partial solution by adding un-visited destinations at the minimum

TABLE V
RMSEL ALGORITHM

INPUT *P, T, Q, b*
OUTPUT $\{\langle p, m, C\rangle\}, cost$

1: $cost \leftarrow \infty$
2: **for each** $p \in P$ **do**
3: **for each** $q \in Q$ **do**
4: $q_{aux} \leftarrow q$
5: $q.p \leftarrow q.p \cup p$
6: $q.m \leftarrow$ findMF(q,b,M)
7: **if** $q.m \neq q_{aux}.m$ **then** $newMF \leftarrow 1$ **else** $newMF \leftarrow 0$
8: $q.C \leftarrow$ computeAvailableSlots(q,b)
9: **if** $q_{aux}.C \cap q.C = \varnothing$ **then** $newST \leftarrow 1$ **else** $newST \leftarrow 0$
10: $nhops \leftarrow$ computeNewHopsInTree(p, q)
11: $ndests \leftarrow$ computeIntermediateDestinations(p, T)
12: $costIte \leftarrow cn^{*}newST + cm^{*}newMF + ch^{*}nhops$ - $ct^{*}ndests$-$cs^{*}|q.C|$
13: **if** $cost < costIte$ **then**
14: $cost \leftarrow costIte$
15: $Q' \leftarrow Q$
16: **if** $newST = 1$ **then** $Q' \leftarrow Q'U\{\langle p, m, C\rangle\}$
17: **else** update $(Q',\ q_{aux}, q)$
18: **return** $Q' = \{\langle p, m, C\rangle\}, cost$

cost. The RMSel algorithm receives among other a set P of pre-computed paths from the source to one destination in T evaluated over the optical layer. The inclusion of every p in P to a tree in Q is evaluated to find the path with the minimum cost according to a cost function. For very path and tree combination, the path is merged with the tree (line 5 in Table V). Since the length of the longest source-destination pair in the tree could be extended beyond the current reach, the modulation format is recomputed (line 6); *newMF* variable is activated when the modulation format has been modified (line 7). Next, the set of end-to-end available slots for the tree are computed (line 8) and compared to the set of slots available before path merging (lines 8–9). Note that a null intersection means that no merging with the tree is possible and therefore, a new tree would be created.

The selected path can entail increasing the number of hops of the tree and also visiting additional intermediate destinations along the path; both variables are stored for further cost computation (lines 10–11). The cost function consists in non-negative coefficients and already computed variables (line 12); the cost is minimized when: *i*) no new tree is required; *ii*) no modulation format change of an existing tree is required; *iii*) the path adds few new hops to an existing tree; *iv*) the number of intermediate destinations in the path increases; and *v*) the number of available slots increases. In case the cost is minimized, the best solution is generated (lines 13–17).

Assuming the worst case, where only one destination is added at each iteration, the complexity of the algorithm in Table IV is $O(|T|^2/2)$. Note that since the RMSel algorithm basically consists in evaluating costs of already pre-computed routes, its complexity is low.

IV. In-Operation Network Re-Optimization

As an example of in-operation network re-optimization, in this section we focus on spectrum defragmentation. In dynamic scenarios, spectrum fragmentation appears in EONs, where after establishing and releasing connection, the optical spectrum is divided into small fragments; this makes difficult finding contiguous spectrum of the required width for incoming connection requests. Spectrum fragmentation increases thus, the blocking probability of connection requests.

To improve network efficiency, defragmentation can be applied by re-configuring selected lightpaths to compact the utilized resources thus, facilitating that incoming connection requests can be served.

Several defragmentation strategies (including rerouting and spectrum reallocation) have been proposed so far for optical networks in general and can be applied to the specific case of EONs. These strategies can be divided into *proactive* and *reactive* considering the way they are triggered [24]. The proactive strategy focuses on minimizing fragmentation itself and it is usually run periodically. Note that this strategy requires long computation times as a result of the amount of data to be processed and hence, it is usually performed during low activity periods, e.g. during nights [25].

In contrast, the *reactive* strategy, also known as *path-triggered*, focuses on making enough room for a given connection request in case that the provisioning algorithm cannot find a feasible resource allocation. This strategy involves only a limited set of already established connections and thus, it might provide solutions in shorter times and can be run in real time.

In this section, we first review a spectrum re-allocation algorithm triggered to reallocate already established lightpaths before an incoming connection request is blocked. Because spectrum re-allocation might cause traffic disruption a procedure for hitless spectrum defragmentation can be applied to shift the central frequency of already established lightpaths to create wider contiguous spectrum fragments to be allocated to the incoming request. The spectrum shifting algorithm is eventually extended to facilitate elastic spectrum operations.

A. Spectrum Reallocation

Aiming at improving the blocking probability in dynamic EONs, authors in [26] proposed a mechanism called *Spectrum Reallocation* (SPRESSO) to reallocate already established connections in the spectrum without rerouting them, so as to make enough room for the incoming connection request.

SPRESSO follows a path-triggered scheme, where the algorithm is triggered whenever not enough resources have been found for an incoming connection request. Every link in the path from a set of *k*-shortest paths connecting source and destination nodes is checked to know whether the amount of available frequency slices is enough for the incoming connection request using a modulation format compatible with the selected path. If enough frequency slices are available in one of the shortest paths, the SPRESSO mechanism is run to find a set of reallocations of already established lightpaths that make enough room for the connection request. Otherwise, the connection request is blocked. Fig. 4 summarizes the routing and spectrum (re)allocation algorithm.

The Spectrum Reallocation problem can be formally stated as follows:

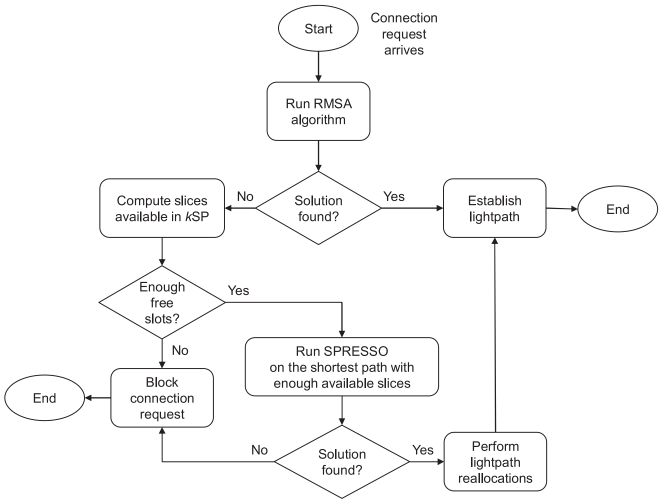


Fig. 4. Algorithm for routing and spectrum (re)allocation.

TABLE VI
ALGORITHM FOR THE SPRING PROBLEM

INPUT E, p, n_{req}
OUTPUT *Solution*

1: $Sol \leftarrow \emptyset$
2: $minLPaths \leftarrow INFINITE$
3: **for** $s = 1..|S|$ **do**
4: $L^+ \leftarrow \varnothing; L^- \leftarrow \varnothing; L^s \leftarrow \varnothing$
5: **for each** $e \in p$ **do**
6: **if** $l(e, s-1) \neq l(e, s)$ **then**
7: $L^- \leftarrow L^- U\{l(e, s^-)\}$
8: $L^+ \leftarrow L^+ U\{l(e, s^+)\}$
9: **else**
10: $L^s \leftarrow L^s U\{l(e, s)\}$
11: $<n_{max}, numLPaths> \leftarrow$ getMaxShift(L^-, L^s, L^+, s)
12: **if** $n_{max} \geq n_{req}$ AND $numLPaths < minLPaths$ **then**
13: $Sol \leftarrow <L^-, L^s, L^+, s>$
14: $minLPaths \leftarrow numLPaths$
15: **return** *Sol*

Given:

1) a connected graph G(N, E);
2) the availability of every frequency slice in the optical spectrum of every link in E and a set M of modulation formats;
3) a connection request to be served defined by the tuple $\langle o, t, b\rangle$. A path p for the request has been already selected but there is no feasible spectrum allocation. On path p, modulation m provides the best spectrum efficiency, so n_{req} slices are needed;
4) a set L of lightpaths already established on the network, where each lightpath l is defined by the tuple $\langle p_l, f_l, n_l\rangle$;
5) the maximum number of lightpaths to be reallocated.

Output: the new central frequency f_l' for the lightpaths to be reallocated and the spectrum allocation $\langle f, n\rangle$ for the request.

Objective: Minimize the amount of paths to be reallocated so to serve the connection request.

B. Spectrum Shifting

Reallocating lightpaths might cause traffic disruption. Taking advantage of the push-pull technique, authors in [27] proposed using it for defragmentation and experimentally demonstrated the spectrum shifting (SPRING) problem.

The heuristic algorithm described in Table VI can be used to solve the SPRING problem; the algorithm iterates on every frequency slice s in the spectrum to find the set of lightpaths using links in path p that are allocated using the closest slice with index lower than $s(s^-)$, the set of lightpaths allocated using the closest slice with index greater than $s(s^+)$ and the set of lightpaths allocated using s (lines 3–10 in Table VI).

The procedure *getMaxShift* finds the widest slot that can be generated by shifting lightpaths (line 11). Lightpaths in set P^- are left shifted, lightpaths in P^+ are right shifted, and lightpaths in P^s are shifted left and right and the option generating the widest slot is chosen. If a slot with at least n_{req} contiguous slices by shifting lightpaths is found and the set of lightpaths involved is lower than that of the best solution found so far, the set of lightpaths is stored as the best solution and the number of lightpaths involved is updated (lines 12–14). The best solution found is eventually returned.

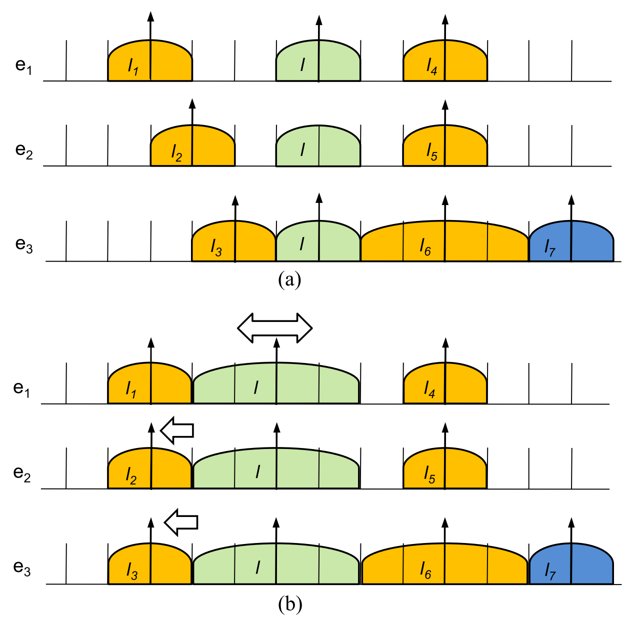


Fig. 5. Example of elastic provisioning with defragmentation.

The complexity of the SPRING algorithm is $O(|S| \cdot |E|)$ for the selected path and as a result, it can be run in short times.

C. Time-Varying Traffic With Defragmentation

As already stated, one of the main features of EONs is their inherent capacity to allocate and dynamically modify the spectrum allocated to lightpaths. However, some of the required frequency slices might be allocated to other lightpaths, which would prevent the operation to be eventually implemented and some traffic would be rejected. A solution to the above problem is to shift neighboring lightpaths to allow the elastic operation to be performed.

For illustrative purposes, Fig. 5 shows a scenario with a number of already established lightpaths. An elastic operation

TABLE VII
ELASTICITY AND SHIFTING ALGORITHM

INPUT $G, L, l = \langle p, m, f, n \rangle, b_{req}$
OUTPUT *Solution*

1: $\langle f', n' \rangle \leftarrow$ ElasticSpectrumAlloc (G, L, l, b_{req})
2: **if** $n' = n_{req}$ **then**
3:     **return** $<\emptyset$, realloc $(l, f', n')>$
4: $L^- \leftarrow \emptyset, L^+ \leftarrow \emptyset$
5: **for each** $e \in p$ **do**
6:     $L^- \leftarrow L^- U\{l' \in L: e \in p_l, \; adjacents\,(l, l'), \; f_l < f\}$
7:     $L^+ \leftarrow L^+ \cup \{l' \in L: e \in p_{l'}, adjacents(l, l'), f_l > f\}$
8: $<n_{max}$, *numLPaths*$> \leftarrow$ getMaxShift$(L^-, \emptyset, L^+, f)$
9: **if** $n_{max} > n$ **then**
10:     $M \leftarrow$ findShifting(L^-, L^+, l, n_{max})
11:     $<f', n'> \leftarrow$ findNewSlot(l, n_{max}, M)
12:     **return** $<M$, realloc$(l, f', n')>$
13: **return** $\emptyset$

is requested to increase the spectrum allocation of lightpath $l = \langle p, m, f, n \rangle$ in two slices (n_{req}). As shown in Fig. 5(a), the request cannot be fulfilled since other established lightpaths in the path *p* are currently using neighboring slices. Note, however, that enough free spectrum to fulfill *l* requirement exist; the central frequency of some connections can be shifted thus, releasing frequency slices that can be used to increase *n* to n_{req}. For instance, the central frequency of lightpaths l_2 and l_3 can to be shifted to make enough room for *l* elastic request Fig. 5(b). Once paths l_2 and l_3 have been shifted, the spectrum allocated to *l* can be increased.

Authors in [28] extended further their defragmentation algorithm, to be run not only to make enough room for new connection requests, but also when elastic operations over already established connections are requested. The problem can be formally stated as:

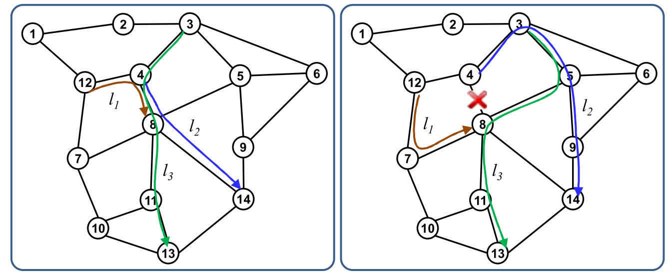
Given:

1) a connected graph G(*N*, *E*);
2) the availability of every frequency slice in the optical spectrum of every link in *E*;
3) a set *L* of lightpaths already established on the network, where each *l* is defined by the tuple $\langle p_l, m_l, f_l, n_l \rangle$;
4) a lightpath *l* in *L* for which a bandwidth adaptation request arrives to increase the bitrate to b_{req} that translates into a n_{req} slices.

Output*:* the list of lightpaths to be shifted and the new values for the spectrum allocation of the given lightpath $\langle f', n' \rangle$.

Objective: minimize the amount of unserved bitrate and the number of shifting operations to be performed.

Table VII presents the proposed algorithm, which combines in a single algorithm both, SPRING and the algorithm described in Table III for elastic provisioning, to perform elastic provisioning with defragmentation. The algorithm first tries to find a new slot $\langle f', n'_{req} \rangle$ for lightpath *l* (line 1). If a new slot of the requested width is found without shifting already established lightpaths, the elastic operation is returned (lines 2–3). In case that no suitable slot is found, the defragmentation process is triggered (lines 4–12). Similarly to the algorithm in Table III, among all the lightpaths whose path p_l share links with *p*, only those with spectrum allocation adjacent to *l* are selected and the maximum shifting of every adjacent lightpath in terms of slices, is computed (lines 5–7). Next, the *getMaxShift* procedure described in Section IV-B is used to compute the maximum slot that can be generated with the minimum number of lightpaths to be shifted (line 8). If any feasible shifting can be done to increase the current slot width, the list of shiftings to be performed *M* is computed (line 10) and a new slot is found taking into account the shiftings in *M* (line 11) and the solution is eventually returned (line 12).



(a) (b)

Fig. 6. Example network (a) before a link failure and (b) after restoration.

Once existing algorithms for provisioning and defragmentation have been reviewed, the next two sections present two use cases that extend the concepts presented in this and the last section.

V. USE CASE I: MODULATION FORMAT-AWARE RESTORATION AND RE-OPTIMIZATION

As a use case of RMSA, let us now to illustrate how the algorithm in Table II can be applied in the case of a failure to find restoration paths for the affected lightpaths [29]. In this use case, we assume that when a failure occurs, an RMSA-based restoration algorithm computes a global solution for all the affected lightpaths. However, as a result of the combined effect of longer routes and lack of resources, the bitrate of the restored lightpaths could become squeezed. Consequently, when the failure is repaired, an in-operation planning algorithm can be run to compute shortest paths that allow, not only to save resources as proposed in [30], but also to increase the bitrate of those lightpaths which bitrate was squeezed.

Modulation format-aware recovery and re-optimization algorithms must select the appropriate modulation format, among those available in the installed TPs, for the lightpaths being restored or re-optimized, respectively. An example of such modulation format awareness is illustrated in Fig. 6 for the depicted 14-node Spanish topology; three lightpaths are established in Fig. 6(a) and they are restored in Fig. 6(b) in the event of a failure in link 4–8. Table VIII shows the resulting restored lightpaths characteristics, where main changes are highlighted. Three different cases can be identified as a result of longer restoration routes (in km) and/or resources availability: *i*) the lower spectral efficiency of the new modulation format entailed a wider slot to serve all the requested bitrate (l_1); *ii*) although the slot width was kept invariant the restored bitrate was squeezed (l_2); and *iii*) the allocated slot was narrower and the bitrate squeezed (l_3).

TABLE VIII
LSP CONFIGURATION

Light path	BitRate(Gb/s)	Len.(km)	Hops	Mod.Format	# Slices
Before Failure					
l_1	200	450	2	DP-QAM16	2
l_2	400	650	2	DP-QAM16	4
l_3	400	1000	4	DP-QAM8	6
After Restoration					
l_1	200	1000	2	DP-QAM8	4
l_2	300	1300	4	DP-QAM8	4
l_3	200	1900	4	DP-QPSK	4

TABLE IX
MF_RESTORATION HEURISTIC ALGORITHM

IN: *G*(*N*, *E*), E_F, *D*
OUT: *Solution*

```
1:  bestSol ← Ø
2:  for each d in D do deallocate(G, d.l)
3:  E ← E\E_F
4:  for i = 1..maxIter do
5:      Sol ← Ø; G' ← G
6:      sort(D, random)
7:      for each d in D do
8:          d.l = ⟨p_d, m_d, c_d⟩ ← RMSA(G', d, minSlotWidth)
9:          if p_d = Ø then continue
10:         allocate (G', d.l)
11:         Sol ← Sol ∪ {d}
12:     sort (Sol, d.b-bitrate(d.l), DESC)
13:     for each d in Sol do
14:         if d.b < bitrate(d.l) then d.l ← expandSlot(G', d)
15:     if Φ(Sol) > Φ(bestSol) then bestSol ← Sol
16: return bestSol
```

It is clear that when link 4–8 is repaired, re-optimization in the network can be performed not only to improve resource utilization by using shorter paths in terms of hops, but also to use more efficient modulation formats and to serve the bitrate originally requested.

The MF-RESTORATION problem can be stated as follows:

Given:

1) a connected graph G(*N*, *E*);
2) the availability of every frequency slice in the optical spectrum of every link in *E* and a set *M* of modulation formats;
3) a set E_F of failed links;
4) a set *D* of demands affected by the failure to be restored. Every demand *d* is defined by the tuple $\langle o_d, t_d, b_d \rangle$.

Output: the lightpath $\langle p_d, m_d, c_d \rangle$ for every demand *d* in *D*.

Objective: minimize the amount of used optical resources while minimizing the total unserved bitrate.

To solve the MF-RESTORATION problem, we propose the heuristic algorithm presented in Table IX. The algorithm first de-allocates the demands in *D* (line 2 in Table IX) and removes the failed links from G (line 3). Next, *maxIter* solutions are generated, each allocating the demands in *D* in a random order (lines 4–16). At each iteration, a new solution *Sol* is created, where failed demands are firstly restored using the minimum slot width (lines 7–12) and, in next, the allocated slot is expanded to increment the restored bitrate (lines 13–14). The fitness of the generated solution *S* is computed and it is stored in case it improves the best solution found so far (*bestSol*) (line 15). The best solution is eventually returned.

TABLE X
MF_AFRO HEURISTIC ALGORITHM

IN: *G*(*N*, *E*), *L*
OUT: *Solution*

```
1:  D ← L.d; bestSol ← D
2:  for each l in L do deallocate(G, l)
3:  for i = 1..maxIter do
4:      Sol ← Ø; G' ← G; feasible ← true
5:      sort(D, random)
6:      for each d in D do
7:          d.l' =< p'_d, m'_d, c'_d > ← RMSA(G', <o_d, t_d, bitrate(d.l)>, ∞)
8:      if p'_d = Ø OR bitrate(d.l')<bitrate(d.l) then
9:          feasible ← false; break
10:     allocate(G', d.l'); Sol ← Sol U {d}
11:  if NOT feasible then continue
12:  for each d in S do
13:      if d.b < bitrate(d.l') then d.l' ← expandSlot(G', d)
14:  if Φ(Sol) < Φ(bestSol) then bestSol ← Sol
15: return bestSol
```

When failed links are repaired, re-optimization of the network can be triggered targeting at improving resource utilization by using shortest paths that include the repaired links and at increasing the bitrate of those demands whose bitrate was squeezed during the restoration. The modulation format aware after failure repair optimization (MF-AFRO) problem can be formally stated as follows:

Given:

1) a connected graph G(*N*, *E*);
2) the availability of every frequency slice in the optical spectrum of every link in *E* and a set *M* of modulation formats;
3) a set *L* of lightpaths candidate for re-optimization, where every lightpath *l* serves a different demand *d*, which specifies the bitrate originally requested.

Output: the re-optimization lightpath $\langle p'_d, m'_d, c'_d \rangle$ for every demand *d*.

Objective: minimize the total unserved bitrate whilst minimizing the amount of used optical resources.

To solve the MF-AFRO, we propose a similar approach to that for the MF-RESTORATION problem; the heuristic algorithm is presented in Table X. In this algorithm we assume that graph G is updated with the current link availability. The algorithm finds a feasible lightpath for each of the demands that can convey, at least, the currently served bitrate. Once a feasible (hopefully shorter) lightpath is found, the selected slot is tried to be expanded to serve as much bitrate as possible targeting at serving the originally requested bitrate.

VI. USE CASE II: VIRTUAL NETWORK RECONFIGURATION

This section presents a second use case of re-optimization, this case in multilayer IP/MPLS transport networks triggered by

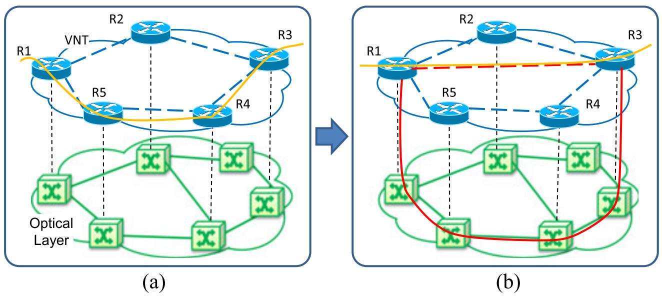


Fig. 7. (a) Initial VNT and OD 1→3 routing. (b) Reconfigured VNT and OD pair 1→3 routing after traffic anomaly detection.

the detection of an anomalous amount of traffic between a source and a destination node. In such networks, VNTs are created by connecting routers through virtual links (*vlinks*) supported by lightpaths in the optical layer; traffic flows are conveyed through IP/MPLS paths along vlinks on the VNT.

An interesting topic is that of the VNT reconfiguration, where TPs installed in routers can be used to increase the capacity of existing vlinks or even support new vlinks so as to follow traffic variations [31]. To adapt the VNT to the current traffic, the capacity of an existing vlink can be increased by setting-up a parallel lightpath when the volume of traffic through that vlink reaches some threshold (e.g., 90% of its capacity). The threshold value must be configured in a way that traffic increments can be conveyed with the remaining capacity while the new lightpath is being established (e.g., some minutes). However, when the traffic experiments an abrupt increment in a short period of time vlinks capacity might be exceeded and thus, traffic losses appear until the new lightpath becomes available. In consequence, a prompt detection of *traffic anomalies* becomes essential to trigger VNT reconfiguration so as to anticipate the capacity for the traffic increment [32].

An example is shown in Fig. 7, where OD traffic is monitored (see Fig. 7(a)) and a traffic anomaly in OD pair 1→3 is detected. The VNT is reconfigured by creating new vlink 1→3 and traffic is rerouted (see Fig. 7(b)). As a result of the anticipated VNT reconfiguration, traffic losses might be decreased.

Based on the above, the OD traffic anomaly-triggered topology reconfiguration (ODEON) problem can be formally stated as:

Given:

1) a graph $G(V, A)$, where V is the set of IP/MPLS routers and A is the set of directed vlinks connecting two routers. The remaining capacity of each vlink e (b_e) is also known;
2) a set $TP(v)$ of TPs installed in each router v. Some of these TPs can be currently used for the existing vlinks and some of them might be unused;
3) the OD pair d affected by the anomaly; d is defined by the tuple $\langle o, t, b \rangle$, where o and t are the source and target routers, respectively, and b is the maximum expected bitrate during anomaly lifetime.

Output: The capacity increments in existing vlinks and the new vlinks to be created so as to serve d. In addition, the new path for d needs to be specified.

TABLE XI
ODEON Algorithm

IN: $G(V, A)$, $d = <o,t,b>$
OUT: $<L, p>$

```
1:  deallocate(G, d)
2:  for each a in A do
3:      if availableCapacity(a) ≤ b then A ← A \ {a}
4:  G'(V, A') ← augment(G)
5:  p ← SP(G', <o, t>)
6:  if p = Ø then return INFEASIBLE
7:  L ← Ø
8:  for each a in p do
9:      if a = (v1 ,v2 ) ∈ A'\A then
10:         l ← RMSA(G', <v1 , v2 , b>, ∞)
11:         L ← L U {l}
12: return <L, p>
```

TABLE XII
Improvement Using Odeon

Max OD Traffic (Gb/s)	Loss (Gb) without ODEON	Loss (Gb) with ODEON	Loss reduction (%)			
			avg	1 a.m.	9 a.m.	5 p.m.
40	15	10	33%	9%	22%	20%
60	20	13	33%	58%	13%	56%
80	28	18	38%	21%	18%	54%
100	45	28	37%	43%	17%	54%
120	49	21	56%	40%	36%	76%
140	48	27	44%	29%	18%	80%

Objective: Minimize the use of resources to serve d, including TPs utilization.

To tackle the ODEON problem, we propose the algorithm in Table XI. The resources currently allocated to pair d are first released (line 1 in Table XI) and the vlinks without enough capacity are removed from the graph (lines 2–3). Next, the graph is augmented where new vlinks are created between routers with available TPs (line 4) and a shortest path is computed (line 5), assuming vlinks cost as follows:

$$c_a = \begin{cases} 1 & \mathit{if}\ a \in A \\ |A| + 1 & \mathit{otherwise}\ (i.e.,\ \mathit{if}\ e \in A' \backslash A). \end{cases} \tag{19}$$

Finally, the RMSA problem is solved for every new vlink and for existing vlinks where the capacity need to be increased. Note that for the latters, an elastic spectrum allocation to increment the capacity of the underlying lightpaths (algorithm in Table III) could be performed instead.

For evaluation purposes, we developed an ad-hoc event driven simulator to evaluate the performance of the proposed ODEON algorithm that reconfigures the network in case of a sudden anomaly. Lightpath set-up path was set to 1 min. The gain in terms of traffic loss is analysed for different values of maximum traffic expected per OD.

Table XII shows the results of traffic loss with and without ODEON assuming that an anomaly can occur at any hour, as well as the loss reduction at three different day times. In view of the results, we can conclude that ODEON reduces traffic losses

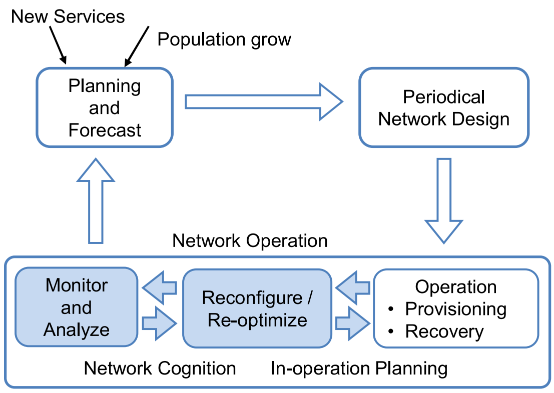


Fig. 8. Extended network life-cycle.

in more than 30% in average, reaching up to 80% of reduction for specific hours and loads.

VII. Concluding Remarks

Technology behind EONs is becoming ready and the features it enables will be available in operators' core transport networks in the near future. Among those features, the capacity to create optical connections with different bitrate and beyond 100 Gb/s and to adapt the spectrum allocation are of great interest for network operators, since it opens providing new services and the opportunity to improve the way transport networks are currently design and operated.

This paper reviewed optimization problems targeting at the design, operation, and re-optimization of EONs. Since most of those problems consist in solving the RMSA problem, two different ILP formulations (the link-path and the node-link) to model that problem were firstly introduced focused on off-line planning problems, were a set of lightpaths need to be served. Because one application of EONs is for datacenter interconnection, a node-link -based ILP formulation for light-trees was eventually presented.

When an EON is in operation, provisioning algorithms are needed to compute in real time the lightpath or the light-tree of incoming connection requests. In addition, one of the new features of EONs is their capability to adapt the spectrum allocation to follow changes in the traffic. Algorithms for these three types of problems related to provisioning were reviewed.

In such dynamic scenarios, spectrum fragmentation has been reported to increase blocking probability in EONs. To mitigate such problem, defragmentation based on spectrum re-allocation and on spectrum shifting has been reviewed to be triggered before a connection request or an elastic spectrum operation could not be performed as a result of a fragmented spectrum. This type of re-optimization opens opportunities to improve resource utilization in EONs and was named as in-operation network planning.

Illustrative use cases were afterwards presented. Starting from a new type of restoration that recovers part of the bitrate of the connections affected by a failure, we propose to apply in-operation planning after the cause of failure has been repaired to increase, if possible, part of the unrestored bitrate.

Another use case of re-configuration is in the case of a traffic anomaly, which entails analyzing monitoring data to anticipate traffic congestion; anomaly detection is used to trigger lightpath provisioning. This example illustrates how monitoring-based data analytics together with in-operation planning derives in adding cognition to the network. In fact, we are extending the classical network life-cycle presented in Section I (see Fig. 8) so new methodologies to improve network resources and thus, reduce costs can be devised. Note that the life-cycle can be further extended to consider on-demand incremental capacity planning [33] as an intermediate way to add capacity between planning cycles.

References

[1] L. Velasco and M. Ruiz, "Design and re-optimization algorithms for elastic optical networks," presented at the Invited Tutorial IEEE/OSA Optical Fiber Communication Conf., Anaheim, CA, USA, 2016.
[2] V. López and L. Velasco, *Elastic Optical Networks: Architectures, Technologies, and Control* (Optical Networks Book Series). Berlin, Germany: Springer-Verlag, 2016.
[3] A. Napoli *et al.*, "Next generation elastic optical networks: The vision of the European research project idealist," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 152–162, Feb. 2015.
[4] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas,"Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 1, pp. 26–38, Jan. 2013.
[5] L. Velasco *et al.*, "Cross-stratum orchestration and flexgrid optical networks for datacenter federations," *IEEE Netw. Mag.*, vol. 27, no. 6, pp. 23–30, Nov./Dec. 2013.
[6] A. Asensio and L. Velasco, "Managing transfer-based datacenter connections," *J. Opt. Commun. Netw.*, vol. 6, no. 7, pp. 660–669, Jul. 2014.
[7] L. Velasco *et al.*, "A service-oriented hybrid access network and cloud architecture," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 159–165, Apr. 2015.
[8] M. Ruiz *et al.*, "Planning fixed to flexgrid gradual migration: Drivers and open issues," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 70–76, Jan. 2014.
[9] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, "In-operation network planning," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 52–60, Jan. 2014.
[10] L. Velasco, A. Castro, M. Ruiz, and G. Junyent, "Solving routing and spectrum allocation related optimization problems: From off-line to in-operation flexgrid network planning (invited tutorial)," *J. Lightw. Technol.*, vol. 32, no. 16, pp. 2780–2795, Aug. 2014.
[11] M. Ruiz, M. Pióro, M. Zotkiewicz, M. Klinkowski, and L. Velasco, "Column generation algorithm for RSA problems in flexgrid optical networks," *Photonic Netw. Commun.*, vol. 26, pp. 53–64, 2013.
[12] M. Klinkowski, M. Zotkiewicz, K. Walkowiak, M. Pióro, M. Ruiz, and L. Velasco, "Solving large instances of the RSA problem in flexgrid elastic optical networks," *J. Opt. Commun. Netw.*, vol. 8, no. 5, pp. 320–330, May 2016.
[13] O. Pedrola, A. Castro, L. Velasco, M. Ruiz, J. P. Fernández-Palacios, and D. Careglio, "CAPEX study for multilayer IP/MPLS over flexgrid optical network," *J. Opt. Commun. Netw.*, vol. 4, no. 8, pp. 639–650, Aug. 2012.
[14] L. Velasco, P. Wright, A. Lord, and G. Junyent, "Saving CAPEX by extending flexgrid-based core optical networks towards the edges (invited paper)," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A171–A183, Oct. 2013.
[15] M. Ruiz and L. Velasco, "Performance evaluation of light-tree schemes in flexgrid optical networks," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1731–1734, Oct. 2014.
[16] M. Charikar *et al.*, "Approximation algorithms for directed Steiner problems," *J. Algorithms*, vol. 33, pp. 73–91, 1999.
[17] M. Dallaglio, A. Giorgetti, N. Sambo, L. Velasco, and P. Castoldi, "Routing, spectrum, and transponder assignment (RSTA) in elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 22, pp. 4648–4658, Nov. 2015.
[18] O. González de Dios *et al.*, "Experimental demonstration of multi-vendor and multi-domain elastic optical network with data and control interoperability over a pan-European test-bed," *J. Lightw. Technol.*, vol. 34, no. 7, pp. 1610–1617, Apr. 2016.

[19] E. Martins and M. Pascoal, “A new implementation of Yen’s ranking loopless paths algorithm,” *4OR, Quart. J. Oper. Res.*, vol. 1, no. 2, pp. 121–133, 2003.

[20] A. Asensio *et al.*, “Impact of aggregation level on the performance of dynamic lightpath adaptation under time-varying traffic,” in *Proc. IEEE 17th Int. Conf. Opt. Netw. Des. Model.*, 2013, pp. 184–189.

[21] F. Cugini *et al.*, “Push-pull defragmentation without traffic disruption in flexible grid optical networks,” *J. Lightw. Technol.*, vol. 31, no. 1, pp. 125–133, Jan. 2013.

[22] M. Ruiz and L. Velasco, “Serving multicast requests on single layer and multilayer flexgrid networks,” *J. Opt. Commun. Netw.*, vol. 7, no. 3, pp. 146–155, Mar. 2015.

[23] L. Gifre *et al.*, “Experimental assessment of ABNO-driven multicast connectivity in flexgrid networks (invited paper),” *J. Lightw. Technol.*, vol. 33, no. 8, pp. 1549–1556, Apr. 2015.

[24] R. Wang and B. Mukherjee, “Provisioning in elastic optical networks with non-disruptive defragmentation,” *J. Lightw. Technol.*, vol. 31, no. 15, pp. 2491–2500, Aug. 2013.

[25] F. Agraz *et al.*, “Design and implementation of a GMPLS-controlled grooming-capable optical transport network,” *J. Opt. Commun. Netw.*, vol. 1, no. 2, pp. A258–A269, Jul. 2009.

[26] A. Castro, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio, “Dynamic routing and spectrum (re)allocation in future flexgrid optical networks,” *Comput. Netw.*, vol. 56, pp. 2869–2883, 2012.

[27] L. Gifre *et al.*, “Experimental assessment of in-operation spectrum defragmentation (invited paper),” *Photonic Netw. Commun.*, vol. 27, pp. 128–140, 2014.

[28] F. Paolucci *et al.*, “Active PCE demonstration performing elastic operations and hitless defragmentation in flexible grid optical networks,” *Photonic Netw. Commun.*, vol. 29, pp. 57–66, 2015.

[29] A. Castro *et al.*, “Experimental assessment of bulk path restoration in multi-layer networks using PCE-based global concurrent optimization,” *J. Lightw. Technol.*, vol. 32, no. 1, pp. 81–90, Jan. 2014.

[30] M. Zotkiewicz, M. Ruiz, M. Klinkowski, M. Pióro, and L. Velasco, “Reoptimization of dynamic flexgrid optical networks after link failure repairs,” *J. Opt. Commun. Netw.*, vol. 7, no. 1, pp. 49–61, Jan. 2015.

[31] F. Morales, M. Ruiz, and L. Velasco, “Virtual network topology reconfiguration based on big data analytics for traffic prediction,” presented at the IEEE/OSA Optical Fiber Communication Conf., Anaheim, CA, USA, 2016.

[32] A. P. Vela, A. Via, M. Ruiz, and L. Velasco, “Bringing data analytics to the network nodes,” in *Proc. Eur. Conf. Opt. Commun.*, 2016, pp. 1–3, paper Tu.2B2.

[33] L. Velasco, F. Morales, Ll. Gifre, A. Castro, O. González de Dios, and M. Ruiz, “On-demand incremental capacity planning in optical transport networks,” *J. Opt. Commun. Netw.*, vol. 8, no. 1, pp. 11–22, Jan. 2016.

Authors’ biographies not available at the time of publication.